

MBS Real Studio Network Plugin Documentation

Christian Schmitz

May 15, 2012

0.1 Introduction

This is the PDF version of the documentation for the Real Studio Plug-in from Monkeybread Software Germany. Plugin part: MBS Real Studio Network Plugin

0.2 Content

- 1 List of all topics 3
- 2 All items in this plugin 11
- 3 List of all classes 97
- 4 List of all global methods 99

Chapter 1

List of Topics

• 2 Network	11
– 2.1 class TXTRecordMBS	11
* 2.1.1 Bytes as string	12
* 2.1.1 ContainsKey(key as string) as boolean	12
* 2.1.1 ContainsKey(txtRecord as string, key as string) as boolean	13
* 2.1.1 Count as integer	14
* 2.1.1 Count(txtRecord as string) as integer	14
* 2.1.1 KeyAtIndex(index as integer) as string	15
* 2.1.1 KeyAtIndex(txtRecord as string, index as integer) as string	15
* 2.1.1 Length as integer	16
* 2.1.1 RemoveValue(key as string)	16
* 2.1.1 SetValue(key as string, value as string) as integer	17
* 2.1.1 Value(key as string) as string	18
* 2.1.1 Value(txtRecord as string, key as string) as string	18
* 2.1.1 ValueAtIndex(index as integer) as string	19
* 2.1.1 ValueAtIndex(txtRecord as string, index as integer) as string	20
– 2.2 class FTPParseMBS	20
* 2.2.1 Parse(line as string) as boolean	22
* 2.2.2 FlagTryCwd as Integer	22
* 2.2.2 FlagTryRetr as Integer	23
* 2.2.2 ID as String	23
* 2.2.2 IDType as Integer	23
* 2.2.2 ModificationTime as Double	23
* 2.2.2 ModificationTimeType as Integer	24
* 2.2.2 Name as String	24

* 2.2.2 Size as Integer	24
* 2.2.2.SizeType as Integer	25
– 2.7 class DNSServiceResolveMBS	34
* 2.7.1 Resolve(InterfaceIndex as integer, servicename as string, servicetype as string, domain as string) as boolean	35
* 2.7.2 ServiceResolve(flags as integer, InterfaceIndex as integer, ErrorCode as integer, Fullname as string, Hosttarget as string, Port as integer, TxtRecord as string)	36
– 2.3 class PacketSocketMBS	25
* 2.3.1 SendPacket(data as string)	26
* 2.3.1 SendPacket(data as string,code as string)	26
* 2.3.1 SendPacket(data as string,code as string,ID as integer)	26
* 2.3.2 ReceivedPacket(data as string,code as string,ID as integer)	27
– 2.4 Globals	27
* 2.4 ClearOptionsMBS(extends s as SocketCore)	27
* 2.4 DNSAddressToNameIPv6MBS(HostAddress as string) as string	27
* 2.4 DNSAddressToNameMBS(HostAddress as string) as string	27
* 2.4 DNSNameToAddressIPv6MBS(HostName as string) as string	27
* 2.4 DNSNameToAddressMBS(HostName as string) as string	28
* 2.4 OptionMutliCastTTLMBS(extends s as SocketCore) as integer	28
* 2.4 OptionMutliCastTTLMBS(extends s as SocketCore, assigns value as integer)	28
* 2.4 OptionTOSMBS(extends s as SocketCore) as integer	29
* 2.4 OptionTOSMBS(extends s as SocketCore, assigns value as integer)	29
* 2.4 OptionTTLMBS(extends s as SocketCore) as integer	29
* 2.4 OptionTTLMBS(extends s as SocketCore, assigns value as integer)	30
– 2.5 class DNSServiceDomainEnumerationMBS	31
* 2.5.1 EnumerateDomains(Flags as integer, InterfaceIndex as integer) as boolean	31
* 2.5.2 ServiceDomainEnumeration(flags as integer, interfaceIndex as integer, errorcode as integer, Domain as string)	32
– 2.6 class DNSServiceRegisterRecordMBS	33
* 2.6.1 RegisterRecord(Flags as integer, interfaceIndex as integer, fullname as string, rrtype as integer, rrClass as integer, data as string, ttl as integer) as boolean	33
* 2.6.1 UpdateRecord(TXTRecord as string, ttl as integer)	34
* 2.6.2 ServiceRegistration(flags as integer, errorcode as integer)	34
– 2.8 class DNSServiceRegisterMBS	36
* 2.8.1 AddRecord(rrType as integer, TXTRecord as string, ttl as integer)	37
* 2.8.1 Register(Flags as integer, interfaceIndex as integer, servicename as string, servicetype as string, domain as string, host as string, port as integer, txtRecord as string) as boolean	37
* 2.8.1 RemoveRecord	38
* 2.8.1 UpdateRecord(TXTRecord as string, ttl as integer)	38
* 2.8.2 ServiceRegistration(flags as integer, errorcode as integer, Name as string, RegType as string, Domain as string)	38

– 2.9 class WirelessInfoMBS	39
* 2.9.1 client_mode as Integer	39
* 2.9.1 comms_qual as Integer	39
* 2.9.1 link_qual as Integer	39
* 2.9.1 macAddress as String	39
* 2.9.1 name as String	40
* 2.9.1 noise as Integer	40
* 2.9.1 port_stat as Integer	40
* 2.9.1 power as Integer	40
* 2.9.1 signal as Integer	41
* 2.9.1 unknown7 as Integer	41
* 2.9.1 unknown9 as Integer	41
– 2.12 class WindowsEthernetMBS	54
* 2.12.1 Item(index as integer) as WindowsEthernetAdapterMBS	55
* 2.12.1 Update	55
* 2.12.2 Count as Integer	55
– 2.10 class WindowsIPAddressMBS	41
* 2.10.1 IP as String	41
* 2.10.1 Mask as String	42
– 2.11 class WirelessMBS	42
* 2.11.1 BestChannel as integer	43
* 2.11.1 Channels as integer	43
* 2.11.1 Configure	44
* 2.11.1 Enabled as boolean	44
* 2.11.1 Encrypt(inNetworkPassword as string, use104bits as boolean) as string	44
* 2.11.1 GetAssociationInfo as dictionary	44
* 2.11.1 GetCountAdhocList as integer	45
* 2.11.1 GetCountApList as integer	45
* 2.11.1 GetInfoASP as dictionary	45
* 2.11.1 GetItemAdhocList(index as integer) as WirelessNetworkInfoMBS	46
* 2.11.1 GetItemApList(index as integer) as WirelessNetworkInfoMBS	46
* 2.11.1 HCF_GetInfo(RIDno as integer, bufsize as integer) as string	46
* 2.11.1 Info as WirelessInfoMBS	47
* 2.11.1 InterfaceDict as dictionary	47
* 2.11.1 IsAvailable as boolean	47
* 2.11.1 Join(inNetworkName as string)	48
* 2.11.1 Join8021x(SSID as string, Password as string)	48
* 2.11.1 JoinWEP(inNetworkName as string, inNetworkPassword as string)	48
* 2.11.1 MakeIBSS(inNetworkName as string, inNetworkPassword as string, inChannel as integer)	48
* 2.11.1 Power as boolean	49

* 2.11.1 Scan(stripDups as boolean)	49
* 2.11.1 ScanSplit(stripDups as boolean)	50
* 2.11.2 Handle as Integer	50
* 2.11.2 lasterror as Integer	51
* 2.11.3 Internal2Err = -2013261818	51
* 2.11.3 InternalErr = -2013261821	51
* 2.11.3 kWINetworkAdhocFlag = & h0002	51
* 2.11.3 kWINetworkCFPollable = & h0004	51
* 2.11.3 kWINetworkCFPollRequest = & h0008	52
* 2.11.3 kWINetworkChannelAgility = & h0080	52
* 2.11.3 kWINetworkEncryptedFlag = & h0010	52
* 2.11.3 kWINetworkManagedFlag = & h0001	52
* 2.11.3 kWINetworkPBCC = & h0040	52
* 2.11.3 kWINetworkReserved = & hFF00	52
* 2.11.3 kWINetworkShortPreamble = & h0020	53
* 2.11.3 NoIOServiceErr = -2013261822	53
* 2.11.3 NoPowerErr = -2013261813	53
* 2.11.3 OutOfMemErr = -2013261819	53
* 2.11.3 ParamErr = -2013261823	53
* 2.11.3 Unk4Err = -2013261820	53
* 2.11.3 Unk7Err = -2013261817	54
* 2.11.3 Unk8Err = -2013261816	54
* 2.11.3 Unk9Err = -2013261815	54
* 2.11.3 UnkaErr = -2013261814	54
- 2.13 class WindowsEthernetAdapterMBS	55
* 2.13.1 Gateway(index as integer) as WindowsIPAddressMBS	56
* 2.13.1 IP(index as integer) as WindowsIPAddressMBS	56
* 2.13.2 AdapterName as String	56
* 2.13.2 Address as String	56
* 2.13.2 CurrentIpAddress as WindowsIPAddressMBS	57
* 2.13.2 Description as String	57
* 2.13.2 DhcpEnabled as Boolean	57
* 2.13.2 DhcpServer as WindowsIPAddressMBS	57
* 2.13.2 Gatewaycount as Integer	57
* 2.13.2 HaveWins as Boolean	58
* 2.13.2 Index as Integer	58
* 2.13.2 IPcount as Integer	58
* 2.13.2 LeaseExpires as Integer	58
* 2.13.2 LeaseObtained as Integer	58
* 2.13.2 PrimaryWinsServer as WindowsIPAddressMBS	59
* 2.13.2 SecondaryWinsServer as WindowsIPAddressMBS	59

* 2.13.2 Type as Integer	59
– 2.14 class WirelessNetworkInfoMBS	59
* 2.14.1 beaconInterval as Integer	60
* 2.14.1 channel as Integer	60
* 2.14.1 flags as Integer	60
* 2.14.1 macAddress as String	60
* 2.14.1 name as String	60
* 2.14.1 nameLen as Integer	61
* 2.14.1 noise as Integer	61
* 2.14.1 signal as Integer	61
– 2.17 class DNSServiceBaseMBS	74
* 2.17.1 Available as boolean	74
* 2.17.1 Close	74
* 2.17.1 ConstructFullName(Service as string, regtype as string, domain as string) as string	74
* 2.17.1 GetDaemonVersion as integer	75
* 2.17.1 Running as boolean	75
* 2.17.2 Handle as Integer	75
* 2.17.2 Lasterror as Integer	76
* 2.17.3 kClassIN = 1	76
* 2.17.3 kErrorAlreadyRegistered = -65547	76
* 2.17.3 kErrorBadFlags = -65543	76
* 2.17.3 kErrorBadInterfaceIndex = -65552	76
* 2.17.3 kErrorBadParam = -65540	77
* 2.17.3 kErrorBadReference = -65541	77
* 2.17.3 kErrorBadState = -65542	77
* 2.17.3 kErrorBadTime = -65559	77
* 2.17.3 kErrorDoubleNAT = -65558	77
* 2.17.3 kErrorFirewall = -65550	77
* 2.17.3 kErrorIncompatible = -65551	78
* 2.17.3 kErrorInvalid = -65549	78
* 2.17.3 kErrorNameConflict = -65548	78
* 2.17.3 kErrorNATTraversal = -65557	78
* 2.17.3 kErrorNoAuth = -65555	78
* 2.17.3 kErrorNoError = 0	78
* 2.17.3 kErrorNoMemory = -65539	79
* 2.17.3 kErrorNoSuchKey = -65556	79
* 2.17.3 kErrorNoSuchName = -65538	79
* 2.17.3 kErrorNoSuchRecord = -65554	79
* 2.17.3 kErrorNotInitialized = -65545	79
* 2.17.3 kErrorRefused = -65553	79
* 2.17.3 kErrorUnknown = -65537	80

* 2.17.3 kErrorUnsupported = -65544	80
* 2.17.3 kFlagsAdd = 2	80
* 2.17.3 kFlagsAllowRemoteQuery = & h200	80
* 2.17.3 kFlagsBrowseDomains = & h40	80
* 2.17.3 kFlagsDefault = 4	81
* 2.17.3 kFlagsForceMulticast = & h400	81
* 2.17.3 kFlagsLongLivedQuery = & h100	81
* 2.17.3 kFlagsMoreComing = 1	81
* 2.17.3 kFlagsNoAutoRename = 8	81
* 2.17.3 kFlagsRegistrationDomains = & h80	82
* 2.17.3 kFlagsShared = & h10	82
* 2.17.3 kFlagsUnique = & h20	82
* 2.17.3 kInterfaceIndexAny = 0	82
* 2.17.3 kInterfaceIndexLocalOnly = -1	82
* 2.17.3 kMaxDomainName = 1005	83
* 2.17.3 kMaxServiceName = 64	83
* 2.17.3 kTypeA = 1	83
* 2.17.3 kTypeA6 = 38	83
* 2.17.3 kTypeAAAA = 28	83
* 2.17.3 kTypeAFSDB = 18	84
* 2.17.3 kTypeANY = 255	84
* 2.17.3 kTypeATMA = 34	84
* 2.17.3 kTypeAXFR = 252	84
* 2.17.3 kTypeCERT = 37	84
* 2.17.3 kTypeCNAME = 5	85
* 2.17.3 kTypeDNAME = 39	85
* 2.17.3 kTypeEID = 31	85
* 2.17.3 kTypeGPOS = 27	85
* 2.17.3 kTypeHINFO = 13	85
* 2.17.3 kTypeISDN = 20	86
* 2.17.3 kTypeIXFR = 251	86
* 2.17.3 kTypeKEY = 25	86
* 2.17.3 kTypeKX = 36	86
* 2.17.3 kTypeLOC = 29	86
* 2.17.3 kTypeMAILA = 254	87
* 2.17.3 kTypeMAILB = 253	87
* 2.17.3 kTypeMB = 7	87
* 2.17.3 kTypeMD = 3	87
* 2.17.3 kTypeMF = 4	87
* 2.17.3 kTypeMG = 8	88
* 2.17.3 kTypeMINFO = 14	88
* 2.17.3 kTypeMR = 9	88

* 2.17.3 kTypeMX = 15	88
* 2.17.3 kTypeNAPTR = 35	88
* 2.17.3 kTypeNIMLOC = 32	89
* 2.17.3 kTypeNS = 2	89
* 2.17.3 kTypeNSAP = 22	89
* 2.17.3 kTypeNSAP_PTR = 23	89
* 2.17.3 kTypeNULL = 10	89
* 2.17.3 kTypeNXT = 30	90
* 2.17.3 kTypeOPT = 41	90
* 2.17.3 kTypePTR = 12	90
* 2.17.3 kTypePX = 26	90
* 2.17.3 kTypeRP = 17	90
* 2.17.3 kTypeRT = 21	91
* 2.17.3 kTypeSIG = 24	91
* 2.17.3 kTypeSINK = 40	91
* 2.17.3 kTypeSOA = 6	91
* 2.17.3 kTypeSRV = 33	91
* 2.17.3 kTypeTKEY = 249	92
* 2.17.3 kTypeTSIG = 250	92
* 2.17.3 kTypeTXT = 16	92
* 2.17.3 kTypeWKS = 11	92
* 2.17.3 kTypeX25 = 19	92
– 2.15 class DNSServiceBrowseMBS	61
* 2.15.1 Browse(InterfaceIndex as integer, servicetype as string, domain as string) as boolean	62
* 2.15.2 ServiceBrowse(Flags as integer, InterfaceIndex as integer, ErrorCode as integer, ServiceName as string, RegType as string, Domain as string)	63
– 2.16 class DNSLookupMBS	63
* 2.16.1 Addresses(index as integer) as string	64
* 2.16.1 Aliases(index as integer) as string	64
* 2.16.1 FormatIP(ip as string) as string	65
* 2.16.1 LookupHostbyAddress(HostAddressBinary as string) as DNSLookupMBS	65
* 2.16.1 LookupHostbyAddressMT(HostAddressBinary as string) as DNSLookupMBS	66
* 2.16.1 LookupHostbyname(HostName as string) as DNSLookupMBS	67
* 2.16.1 LookupHostbyname(HostName as string, AddressType as integer) as DNSLookupMBS	67
* 2.16.1 LookupHostbynameMT(HostName as string) as DNSLookupMBS	68
* 2.16.1 LookupHostbynameMT(HostName as string, AddressType as integer) as DNSLookupMBS	69
* 2.16.1 ParseIPv4(ip as string) as string	70
* 2.16.1 ParseIPv6(ip as string) as string	70
* 2.16.2 Address as String	71

* 2.16.2	AddressesCount as Integer	71
* 2.16.2	AddressType as Integer	72
* 2.16.2	AliasCount as Integer	72
* 2.16.2	Name as String	73
* 2.16.3	AddressTypeIPv4 = 2	73
* 2.16.3	AddressTypeIPv6 = 10	74
– 2.18	class DarwinPingMBS	93
* 2.18.1	Ping(HostToPing as string, TimeOutMS as integer, TimeToLife as integer) as integer	93
* 2.18.1	SimplePing(HostToPing as string, NumberOfPacketsToSend as integer, PingTimeoutInSeconds as integer, ReturnImmediatelyAfterReply as integer) as integer	93
* 2.18.2	HostToPing as String	94
* 2.18.2	NumberOfPacketsToSend as Integer	94
* 2.18.2	PingTimeoutInSeconds as Integer	94
* 2.18.2	ReturnImmediatelyAfterReply as Integer	94
* 2.18.2	TimeToLife as Integer	94
* 2.18.3	AddressResolved(ip as string)	95
* 2.18.3	Finished(NumberOfPacketsSent as integer, NumberOfPacketsReceived as integer)	95
* 2.18.3	NextPing(PacketSequenceNumber as integer)	95
* 2.18.3	Response(PacketSequenceNumber as integer, ttl as integer, RoundTripTimeInMS as double)	95
* 2.18.3	SentError(PacketSequenceNumber as integer)	95
* 2.18.3	SentSuccess(PacketSequenceNumber as integer)	96
* 2.18.3	Timeout(PacketSequenceNumber as integer)	96

Chapter 2

Network

2.1 class TXTRecordMBS

class TXTRecordMBS

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The class to represent a DNS-SD TXT record.

Example:

```
dim t as new TXTRecordMBS
```

```
call t.SetValue "Hello", "World"
```

```
MsgBox t.Bytes // show complete record
```

```
MsgBox t.Value("Hello") // lookup value
```

Notes: The constructor creates an empty TXTRecord where you can add values using TXTRecordSetValue.

2.1.1 Methods

Bytes as string

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Allows you to retrieve a copy of to the raw bytes within a TXTRecord.

Example:

```
dim t as new TXTRecordMBS

call t.SetValue "Hello", "World"
```

Notes: Returns a string with the raw bytes inside the TXTRecord which you can pass directly to DNSServiceRegisterMBS.Register() or to DNSServiceRegisterMBS.UpdateRecord().

ContainsKey(key as string) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Allows you to determine if a given TXT Record contains a specified key.

Example:

```
dim t as new TXTRecordMBS

call t.SetValue "Hello", "World"

if t.ContainsKey("Hello") then
  MsgBox "Hello found."
else
  MsgBox "Hello not found ->bug."
end if

if t.ContainsKey("hallo") then
  MsgBox "hallo found ->bug."
else
  MsgBox "hallo not found."
end if
```

Notes:

key: An ASCII string containing the key name.

Returns true if the TXT Record contains the specified key. Otherwise, it returns false.
See also:

- 2.1.1 ContainsKey(txtRecord as string, key as string) as boolean

13

ContainsKey(txtRecord as string, key as string) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Allows you to determine if a given TXT Record contains a specified key.

Example:

```
dim t as new TXTRecordMBS

call t.SetValue "Hello", "World"

dim s as string = t.Bytes // copy record to string

// now search in that string:
if TXTRecordMBS.ContainsKey(s, "Hello") then
  MsgBox "Hello found."
else
  MsgBox "Hello not found ->bug."
end if

if TXTRecordMBS.ContainsKey(s, "hallo") then
  MsgBox "hallo found ->bug."
else
  MsgBox "hallo not found."
end if
```

Notes:

txtRecord: The txt record to search in as a string.
key: An ASCII string containing the key name.

Returns true if the TXT Record contains the specified key. Otherwise, it returns false.
See also:

- 2.1.1 ContainsKey(key as string) as boolean

12

Count as integer

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the number of keys stored in the TXT Record.

Example:

```
dim t as new TXTRecordMBS
```

```
call t.SetValue "Hello1", "World1"
```

```
call t.SetValue "Hello2", "World2"
```

```
call t.SetValue "Hello3", "World3"
```

```
MsgBox str(t.Count)
```

See also:

- 2.1.1 Count(txtRecord as string) as integer

14

Count(txtRecord as string) as integer

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Returns the number of keys stored in the TXT Record.

Example:

```
dim t as new TXTRecordMBS
```

```
call t.SetValue "Hello1", "World1"
```

```
call t.SetValue "Hello2", "World2"
```

```
call t.SetValue "Hello3", "World3"
```

```
dim s as string = t.Bytes // copy record
```

```
// now count in that record
```

```
MsgBox str(TXTRecordMBS.Count(s))
```

See also:

- 2.1.1 Count as integer

KeyAtIndex(index as integer) as string

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the key with the given index from the txt record.

Example:

```
dim t as new TXTRecordMBS
call t.SetValue "Hello", "World"
MsgBox t.KeyAtIndex(0) +EndOfLine + t.ValueAtIndex(0)
```

Notes:

index: the index you want in the range from zero to GetCount()-1.

It's also possible to iterate through keys in a TXT record by simply calling KeyAtIndex() repeatedly, beginning with index zero and increasing until lasterror is set to kErrorInvalid.
See also:

- 2.1.1 KeyAtIndex(txtRecord as string, index as integer) as string

KeyAtIndex(txtRecord as string, index as integer) as string

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Returns the key with the given index from the txt record.

Example:

```
dim t as new TXTRecordMBS
call t.SetValue "Hello", "World"
dim s as string = t.Bytes // copy record to string
// now search in that string:
MsgBox TXTRecordMBS.KeyAtIndex(s,0) +EndOfLine + TXTRecordMBS.ValueAtIndex(s,0)
```

Notes:

txtRecord: The txt record to search in as a string.

index: the index you want in the range from zero to GetCount()-1.

It's also possible to iterate through keys in a TXT record by simply calling KeyAtIndex() repeatedly, beginning with index zero and increasing until lasterror is set to kErrorInvalid.

See also:

- 2.1.1 KeyAtIndex(index as integer) as string

15

Length as integer

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Allows you to determine the length of the raw bytes within a TXTRecord.

Example:

```
dim t as new TXTRecordMBS
```

```
call t.SetValue "Hello1", "World1"
```

```
call t.SetValue "Hello2", "World2"
```

```
call t.SetValue "Hello3", "World3"
```

```
MsgBox str(t.Length) // shows 42
```

RemoveValue(key as string)

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Removes a key from a TXTRecordRef. The "key" must be an ASCII string which exists in the TXTRecord.

Example:

```
dim t as new TXTRecordMBS
```

```
call t.SetValue "Hello1", "World1"
```

```
call t.SetValue "Hello2", "World2"
```

```
call t.SetValue "Hello3", "World3"
```

```
t.RemoveValue "Hello2"
```

```
MsgBox t.Bytes
```

Notes:

key: A key name which exists in the TXTRecord.

Lasterror is kErrorNoError on success or kErrorNoSuchKey if the "key" does not exist in the TXTRecord.

SetValue(key as string, value as string) as integer

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Adds a key (optionally with value) to a TXTRecordRef.

Example:

```
dim t as new TXTRecordMBS
```

```
call t.SetValue "Hello1", "World1"
```

```
call t.SetValue "Hello2", "World2"
```

```
call t.SetValue "Hello3", "World3"
```

Notes:

If the "key" already exists in the TXTRecord, then the current value will be replaced with the new value.

Keys may exist in four states with respect to a given TXT record:

Absent (key does not appear at all)

Present with no value ("key" appears alone)

Present with empty value ("key=" appears in TXT record)

Present with non-empty value ("key=value" appears in TXT record)

For more details refer to "Data Syntax for DNS-SD TXT Records" in

<http://files.dns-sd.org/draft-cheshire-dnsext-dns-sd.txt>

key: A string which only contains printable ASCII values (& h20-& h7E), excluding '=' (& h3D). Keys should be 8 characters or less.

value: Any binary value. For values that represent textual data, UTF-8 is STRONGLY recommended.

Returns kErrorNoError on success. Returns kErrorInvalid if the "key" string contains illegal characters. Returns kErrorNoMemory if adding this key would exceed the available storage.

The plugin currently does not create keys with empty value. If you need that, please email us.

Value(key as string) as string

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Allows you to retrieve the value for a given key from a TXT Record.

Example:

```
dim t as new TXTRecordMBS

call t.SetValue "Hello", "World"
MsgBox t.Value("Hello") // lookup value
```

Notes:

key: The ASCII string containing the key name.

Returns an empty string if the key does not exist in this TXT record, or exists with no value (to differentiate between these two cases use ContainsKey()).

See also:

- 2.1.1 Value(txtRecord as string, key as string) as string

18

Value(txtRecord as string, key as string) as string

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Allows you to retrieve the value for a given key from a TXT Record.

Example:

```

dim t as new TXTRecordMBS
call t.SetValue "Hello", "World"

dim s as string = t.bytes
MsgBox TXTRecordMBS.Value(s, "Hello") // lookup value

```

Notes:

txtRecord: Pointer to the received TXT Record bytes.
key: The ASCII string containing the key name.

Returns an empty string if the key does not exist in this TXT record, or exists with no value (to differentiate between these two cases use ContainsKey()).

See also:

- 2.1.1 Value(key as string) as string

18

ValueAtIndex(index as integer) as string

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the key with the given index from the txt record.

Example:

```

dim t as new TXTRecordMBS
call t.SetValue "Hello", "World"

MsgBox t.KeyAtIndex(0) +EndOfLine + t.ValueAtIndex(0)

```

Notes:

index: the index you want in the range from zero to GetCount()-1.

It's also possible to iterate through values in a TXT record by simply calling ValueAtIndex() repeatedly, beginning with index zero and increasing until lasterror is set to kErrorInvalid.

See also:

- 2.1.1 ValueAtIndex(txtRecord as string, index as integer) as string 20

ValueAtIndex(txtRecord as string, index as integer) as string

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Returns the key with the given index from the txt record.

Example:

```
dim t as new TXTRecordMBS
call t.SetValue "Hello", "World"
dim s as string = t.Bytes // copy record to string
// now search in that string:
MsgBox TXTRecordMBS.KeyAtIndex(s,0) +EndOfLine + TXTRecordMBS.ValueAtIndex(s,0)
```

Notes:

txtRecord: The txt record to search in as a string.

index: the index you want in the range from zero to GetCount()-1.

It's also possible to iterate through values in a TXT record by simply calling GetValueAtIndex() repeatedly, beginning with index zero and increasing until lasterror is set to kErrorInvalid.

See also:

- 2.1.1 ValueAtIndex(index as integer) as string 19

2.2 class FTTParseMBS

class FTTParseMBS

Plugin Version: 4.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for parsing FTP LIST responses.

Example:

```

dim f as new FTTParseMBS

if f.Parse("drwxr-xr-x 2 lefregole nobody 4096 May 30 2011 Downloads") then
if f.ModificationTimeType <>0 then

dim d as new date(1970,1,1,0,0,0)

d.TotalSeconds = d.TotalSeconds+ f.ModificationTime

MsgBox f.Name+" has "+str(f.Size)+" bytes: "+d.ShortDate
else
MsgBox f.Name+" has "+str(f.Size)+" bytes."
END IF
end if

```

Notes:

An implementation of the ftpparse code from <<http://cr.yip.to/ftpparse.html>>. Works only for ASCII text.

What is it?

ftpparse is a library for parsing FTP LIST responses.

ftpparse currently understands the LIST output from any UNIX server, Microsoft FTP Service, Windows NT FTP Server, VMS, WFTPD, NetPresenz, NetWare, and MSDOS. It also supports EPLF, a solution to the LIST-parsing mess.

ftpparse parses file modification times into time_t, so you can easily compare them and display them in your favorite format. It lets you know how precise the time_t is: LOCAL meaning exact with known time zone (available from EPLF), REMOTEMINUTE meaning unknown time zone and seconds, or REMOTEDAY meaning unknown time zone and time of day.

To use ftpparse, simply feed each line of LIST output to the ftpparse() routine, along with a pointer to a struct ftpparse. If ftpparse() returns 1, you can find a filename and various other useful information inside the struct ftpparse.

Commercial use of ftpparse is fine, as long as you let me know what programs you're using it in.

2.2.1 Methods

Parse(line as string) as boolean

Plugin Version: 4.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Parses a line.

Example:

```
dim f as new FTPParseMBS

if f.Parse("drwxr-xr-x 2 lefregole nobody 4096 May 30 2011 Downloads") then
if f.ModificationTimeType <>0 then

dim d as new date(1970,1,1,0,0,0)

d.TotalSeconds = d.TotalSeconds+ f.ModificationTime

MsgBox f.Name+" has "+str(f.Size)+" bytes: "+d.ShortDate
else
MsgBox f.Name+" has "+str(f.Size)+" bytes."
END IF
end if
```

Notes:

Returns true on success.

A filename is minimum required for success.

2.2.2 Properties

FlagTryCwd as Integer

Plugin Version: 4.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Flags.

Notes:

0 if cwd is definitely pointless, 1 otherwise.

(Read and Write property)

FlagTryRetr as Integer

Plugin Version: 4.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Flags.

Notes:

0 if retr is definitely pointless, 1 otherwise.

(Read and Write property)

ID as String

Plugin Version: 4.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unique identifier for the file on the FTP server.

Notes: (Read only property)

IDType as Integer

Plugin Version: 4.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The type of identifier.

Notes:

Possible values:

```
const FTTPARSE_ID_UNKNOWN = 0
const FTTPARSE_ID_FULL   = 1
```

(Read and Write property)

ModificationTime as Double

Plugin Version: 4.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The time value in seconds.

Notes:

When a time zone is unknown, it is assumed to be GMT. You may want to use localtime() for LOCAL times, along with an indication that the time is correct in the local time zone, and gmtime() for REMOTE* times.

The functions `ctime()`, `gmtime()` and `localtime()` all take as an argument a time value representing the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Check the "man localtime" command in the Mac OS X Terminal for more details about this time stuff.
(Read and Write property)

ModificationTimeType as Integer

Plugin Version: 4.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The type of modification time.

Notes:

Possible values:

```
const FTTPARSE_MTIME_UNKNOWN    0
const FTTPARSE_MTIME_LOCAL      1  time is correct
const FTTPARSE_MTIME_REMOTEMINUTE 2  time zone and secs are unknown
const FTTPARSE_MTIME_REMOTEDAY  3  time zone and time of day are unknown
```

(Read and Write property)

Name as String

Plugin Version: 4.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The file name.

Notes: (Read only property)

Size as Integer

Plugin Version: 4.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The file size.

Notes: (Read and Write property)

SizeType as Integer

Plugin Version: 4.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The type of size known.

Notes:

Possible values:

```
const FTTPARSE_SIZE_UNKNOWN 0
const FTTPARSE_SIZE_BINARY  1 size is the number of octets in TYPE I
const FTTPARSE_SIZE_ASCII   2 size is the number of octets in TYPE A
```

(Read and Write property)

2.3 class PacketSocketMBS

class PacketSocketMBS

Plugin Version: 8.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An extension to the socket class for easy sending packets through the net.

Notes:

A packet is made like this:

```
+0: Size of data block
+4: 4 byte code string
+8: 4 byte for ID
+12: variable data.
```

The numbers are send in low endian format so this is crossplatform.

This socket subclass can be easily made in Realbasic code itself. I did it for several of my applications, so if you like it decide between making it yourself or using this one which may even be faster, because it is written in C?

This class is only available in RB 4.0 or newer, because it crashes on RB 3.5.

As the strings are transferred as binary data you need to take care that you send text e.g. as UTF8 and

restore this encoding setting on receiving.
Subclass of the TCPSocket class.

2.3.1 Methods

SendPacket(data as string)

Plugin Version: 8.1 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sends a packet.
Notes: Uses ID=0 and code="" as a shortcut for the long SendPacket call.
See also:

- 2.3.1 SendPacket(data as string,code as string) 26
- 2.3.1 SendPacket(data as string,code as string,ID as integer) 26

SendPacket(data as string,code as string)

Plugin Version: 8.1 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sends a packet.
Notes: Uses ID=0 as a shortcut for the long SendPacket call.
See also:

- 2.3.1 SendPacket(data as string) 26
- 2.3.1 SendPacket(data as string,code as string,ID as integer) 26

SendPacket(data as string,code as string,ID as integer)

Plugin Version: 8.1 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sends a packet.
Notes:

code and ID are optional.
Code is a 4 byte string to identify the content.
(e.g. "mess" for a message or "link" for a link.)
See also:

- 2.3.1 SendPacket(data as string) 26
- 2.3.1 SendPacket(data as string,code as string) 26

2.3.2 Events

ReceivedPacket(data as string,code as string,ID as integer)

Plugin Version: 8.1 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A packet was received.

2.4 Globals

ClearOptionsMBS(extends s as **SocketCore**)

Plugin Version: 8.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clears all options.
Notes: Works only on BSD Sockets and if the socket handle value is valid.

DNSAddressToNameIPv6MBS(HostAddress as string) as string

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: Yes, . **Function:** Resolves an IP to it's name.

DNSAddressToNameMBS(HostAddress as string) as string

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Resolves an IP to it's name.
Notes: Only for IPv4.

DNSNameToAddressIPv6MBS(HostName as string) as string

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: Yes, . **Function:** Resolves the IP address for this name.

Example:

```
msgBox dnsnameToAddressIPv6MBS("localhost") + endofLine + dnsnameToAddressMBS("localhost")
```

DNSNameToAddressMBS(HostName as string) as string

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Resolves the IP address for this name.

Example:

```
msgBox dnsnameToAddressIPv6MBS("localhost") + endofLine + dnsnameToAddressMBS("localhost")
```

Notes: Only for IPv4.

OptionMutliCastTTLMBS(extends s as SocketCore) as integer

Plugin Version: 8.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets the multicast time to live value.

Notes: Works only on BSD Sockets and if the socket handle value is valid.

See also:

- 2.4 OptionMutliCastTTLMBS(extends s as SocketCore, assigns value as integer) 28

OptionMutliCastTTLMBS(extends s as SocketCore, assigns value as integer)

Plugin Version: 8.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the multicast time to live value.

Example:

```
dim t as new TCPSocket

// create a socket
t.Port=80
t.Address="www.apple.com"
t.Connect

// wait a second to connect
app.DoEvents 1000
```

```
// shows handle
MsgBox str(t.Handle)

// shows current value
MsgBox str(t.OptionMutliCastTTLMBS)
t.OptionMutliCastTTLMBS=10

// shows new value
MsgBox str(t.OptionMutliCastTTLMBS)
```

Notes: Works only on BSD Sockets and if the socket handle value is valid.
See also:

- 2.4 OptionMutliCastTTLMBS(extends s as SocketCore) as integer 28

OptionTOSMBS(extends s as SocketCore) as integer

Plugin Version: 8.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets the type of service value.

Notes: Works only on BSD Sockets and if the socket handle value is valid.
See also:

- 2.4 OptionTOSMBS(extends s as SocketCore, assigns value as integer) 29

OptionTOSMBS(extends s as SocketCore, assigns value as integer)

Plugin Version: 8.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the type of service value.

Notes: Works only on BSD Sockets and if the socket handle value is valid.
See also:

- 2.4 OptionTOSMBS(extends s as SocketCore) as integer 29

OptionTTLMBS(extends s as SocketCore) as integer

Plugin Version: 8.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets the time to live value.

Notes: Works only on BSD Sockets and if the socket handle value is valid.
See also:

- 2.4 OptionTTLMBS(extends s as SocketCore, assigns value as integer)

30

OptionTTLMBS(extends s as SocketCore, assigns value as integer)

Plugin Version: 8.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the time to live value.

Example:

```
dim t as new TCPSocket

// create a socket
t.Port=80
t.Address="www.apple.com"
t.Connect

// wait a second to connect
app.DoEvents 1000

// shows handle
MsgBox str(t.Handle)

// shows current value
MsgBox str(t.OptionTTLMBS)
t.OptionTTLMBS=10

// shows new value
MsgBox str(t.OptionTTLMBS)
```

Notes: Works only on BSD Sockets and if the socket handle value is valid.
See also:

- 2.4 OptionTTLMBS(extends s as SocketCore) as integer

29

2.5 class DNSServiceDomainEnumerationMBS

class DNSServiceDomainEnumerationMBS

Plugin Version: 5.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** A class for Apple's implementation of ZeroConfig named Bonjour.

Notes:

This class allows you to enumerate domains.

This class requires Mac OS X 10.3 or Windows (with installed Bonjour support and the dnssd.dll) to work. Use DNSServiceDiscoveryBrowseMBS if you want to support Mac OS X 10.2.

common error codes:

kDNSServiceErr_NoError	= 0
kDNSServiceErr_Unknown	= -65537 (= & hFFFE FFFF)
kDNSServiceErr_NoSuchName	= -65538
kDNSServiceErr_NoMemory	= -65539
kDNSServiceErr_BadParam	= -65540
kDNSServiceErr_BadReference	= -65541
kDNSServiceErr_BadState	= -65542
kDNSServiceErr_BadFlags	= -65543
kDNSServiceErr_Unsupported	= -65544
kDNSServiceErr_NotInitialized	= -65545
kDNSServiceErr_AlreadyRegistered	= -65547
kDNSServiceErr_NameConflict	= -65548
kDNSServiceErr_Invalid	= -65549
kDNSServiceErr_Incompatible	= -65551 (client library incompatible with daemon)
kDNSServiceErr_BadinterfaceIndex	= -65552

Subclass of the DNSServiceBaseMBS class.

2.5.1 Methods

EnumerateDomains(Flags as integer, InterfaceIndex as integer) as boolean

Plugin Version: 5.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Asynchronously enumerate domains available for browsing and registration.

Notes:

Note that the names returned are (like all of DNS-SD) UTF-8 strings, and are escaped using standard DNS escaping rules.

A graphical browser displaying a hierarchical tree-structured view should cut the names at the bare dots to yield individual labels, then de-escape each label according to the escaping rules, and then display the resulting UTF-8 text.

Flags for specifying domain enumeration type in `DNSServiceEnumerateDomains`.

`BrowseDomains` enumerates domains recommended for browsing, `RegistrationDomains` enumerates domains recommended for registration:

```
kDNSServiceFlagsBrowseDomains    = 64
kDNSServiceFlagsRegistrationDomains = 128
```

You need to keep this object alive (keep a reference in a window, a module or your app class) so you can receive events.

2.5.2 Events

ServiceDomainEnumeration(flags as integer, interfaceIndex as integer, errorcode as integer, Domain as string)

Plugin Version: 5.0 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Domains changed.

Notes:

Flags uses this flags: `kFlagsAdd`, `kFlagsDefault`.

”Default” applies only to enumeration and is only valid in conjunction with ”Add”

Starting a new query from within this event on the same object can cause problems. Better you create a new instance for another query if you need one.

2.6 class DNSServiceRegisterRecordMBS

class DNSServiceRegisterRecordMBS

Plugin Version: 8.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** A class for Apple's implementation of ZeroConfig named Bonjour.

Notes:

This class is to register a name record with the dns service.
Subclass of the DNSServiceBaseMBS class.

2.6.1 Methods

RegisterRecord(Flags as integer, interfaceIndex as integer, fullname as string, rrtype as integer, rrClass as integer, data as string, ttl as integer) as boolean

Plugin Version: 8.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Registers a record on the DNS system.

Example:

```
const kDNSServiceClass_ IN = 1
const kDNSServiceType_ SRV = 33
const kDNSServiceFlagsShared = & h10
const kDNSServiceFlagsUnique = & h20
```

Notes:

flags: Possible values are kDNSServiceFlagsShared or kDNSServiceFlagsUnique (see flag type definitions for details).

interfaceIndex: If non-zero, specifies the interface on which to register the record (the index for a given interface is determined via the if_ nametoindex() family of calls.) Passing 0 causes the record to be registered on all interfaces. See "Constants for specifying an interface index" for more details.

fullname: The full domain name of the resource record.

rrtype: The numerical type of the resource record (e.g. kDNSServiceType_ PTR, kDNSServiceType_ SRV, etc)

rrclass: The class of the resource record (usually kDNSServiceClass_ IN)

data: the raw rdata, as it is to appear in the DNS record.

ttl: The time to live of the resource record, in seconds. Pass 0 to use a default value.

Lasterror is set. `kDNSServiceErr_NoError` on success (any subsequent, asynchronous errors are delivered to the event), otherwise returns an error code indicating the error that occurred (the event is never invoked)

You need to keep this object alive (keep a reference in a window, a module or your app class) so you can receive events.

UpdateRecord(TXTRecord as string, ttl as integer)

Plugin Version: 8.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Changes the TXT Record.

Notes:

data is the TXT Record composed of Pascal Strings.

ttl = time to live.

Lasterror is set.

2.6.2 Events

ServiceRegistration(flags as integer, errorcode as integer)

Plugin Version: 8.0 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Registration was done.

Notes: Starting a new query from within this event on the same object can cause problems. Better you create a new instance for another query if you need one.

2.7 class DNSServiceResolveMBS

class DNSServiceResolveMBS

Plugin Version: 5.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** A class for Apple's implementation of ZeroConfig named Bonjour.

Notes:

This class allows you to lookup names on the local network using Bonjour.

This class requires Mac OS X 10.3 or Windows (with installed Bonjour support and the dnssd.dll) to work. Use DNSServiceDiscoveryBrowseMBS if you want to support Mac OS X 10.2.

common error codes:

kDNSServiceErr_NoError	= 0
kDNSServiceErr_Unknown	= -65537 (= & hFFFE FFFF)
kDNSServiceErr_NoSuchName	= -65538
kDNSServiceErr_NoMemory	= -65539
kDNSServiceErr_BadParam	= -65540
kDNSServiceErr_BadReference	= -65541
kDNSServiceErr_BadState	= -65542
kDNSServiceErr_BadFlags	= -65543
kDNSServiceErr_Unsupported	= -65544
kDNSServiceErr_NotInitialized	= -65545
kDNSServiceErr_AlreadyRegistered	= -65547
kDNSServiceErr_NameConflict	= -65548
kDNSServiceErr_Invalid	= -65549
kDNSServiceErr_Incompatible	= -65551 (client library incompatible with daemon)
kDNSServiceErr_BadinterfaceIndex	= -65552

Subclass of the DNSServiceBaseMBS class.

2.7.1 Methods

Resolve(InterfaceIndex as integer, servicename as string, servicetype as string, domain as string) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Resolves a named instance of a service to its address, port, and (optionally) a text description.

Notes:

You need to keep this object alive (keep a reference in a window, a module or your app class) so you can receive events.

If you call this method in response to a browse call, please pass the domain string you got in the Service-Browse event and do not replace it with something else.

2.7.2 Events

ServiceResolve(flags as integer, InterfaceIndex as integer, ErrorCode as integer, Fullname as string, Hosttarget as string, Port as integer, TxtRecord as string)

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Resolve got something.

Notes: Starting a new query from within this event on the same object can cause problems. Better you create a new instance for another query if you need one.

2.8 class DNSServiceRegisterMBS

class DNSServiceRegisterMBS

Plugin Version: 5.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** A class for Apple's implementation of ZeroConfig named Bonjour.

Notes:

This class allows you to register names on the local network using Bonjour.

This class requires Mac OS X 10.3 or Windows (with installed Bonjour support and the dnssd.dll) to work. Use DNSServiceDiscoveryBrowseMBS if you want to support Mac OS X 10.2.

common error codes:

kDNSServiceErr_NoError	= 0
kDNSServiceErr_Unknown	= -65537 (= &hFFFE FFFF)
kDNSServiceErr_NoSuchName	= -65538
kDNSServiceErr_NoMemory	= -65539
kDNSServiceErr_BadParam	= -65540
kDNSServiceErr_BadReference	= -65541
kDNSServiceErr_BadState	= -65542
kDNSServiceErr_BadFlags	= -65543
kDNSServiceErr_Unsupported	= -65544
kDNSServiceErr_NotInitialized	= -65545
kDNSServiceErr_AlreadyRegistered	= -65547
kDNSServiceErr_NameConflict	= -65548
kDNSServiceErr_Invalid	= -65549
kDNSServiceErr_Incompatible	= -65551 (client library incompatible with daemon)
kDNSServiceErr_BadinterfaceIndex	= -65552

Subclass of the DNSServiceBaseMBS class.

2.8.1 Methods

AddRecord(rrType as integer, TXTRecord as string, ttl as integer)

Plugin Version: 8.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Add a record to a registered service.

Notes:

Lasterror is set.

The name of the record will be the same as the registered service's name. The record can later be updated or deregistered by UpdateRecord() or RemoveRecord().

rrtype: The type of the record (e.g. kTypeTXT, kTypeSRV, etc)

data: The raw rdata to be contained in the added resource record.

ttl: The time to live of the resource record, in seconds. Pass 0 to use a default value.

Register(Flags as integer, interfaceIndex as integer, servicename as string, servicetype as string, domain as string, host as string, port as integer, txtRecord as string) as boolean

Plugin Version: 5.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Register a service that is discovered via Browse() and Resolve() calls.

Notes:

Flag for specifying renaming behavior on name conflict when registering non-shared records. NoAutorename is only valid if a name is explicitly specified when registering a service (ie the default name is not used.)

```
kDNSServiceFlagsNoAutoRename    = 8,
kDNSServiceFlagsAutoRename      = 0 (i.e. bit not set)
```

You need to keep this object alive (keep a reference in a window, a module or your app class) so you can receive events.

In a perfect application you use DNSServiceDomainEnumerationMBS class to enumerate which domains you

should use.

RemoveRecord

Plugin Version: 8.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Remove a record previously added to a service record set via AddTextRecord.

Notes: Lasterror is set.

UpdateRecord(TXTRecord as string, ttl as integer)

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Changes the TXT Record.

Notes:

data is the TXT Record composed of Pascal Strings.

ttl = time to live.

Lasterror is set.

2.8.2 Events

ServiceRegistration(flags as integer, errorcode as integer, Name as string, RegType as string, Domain as string)

Plugin Version: 5.0 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Registration was done.

Notes: Starting a new query from within this event on the same object can cause problems. Better you create a new instance for another query if you need one.

2.9 class WirelessInfoMBS

class WirelessInfoMBS

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The data structure to hold information about the current state.

2.9.1 Properties

client_mode as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The client mode: 1 = BSS, 4 = Create IBSS

Notes: (Read and Write property)

comms_qual as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Communication Quality

Notes: (Read and Write property)

link_qual as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Link quality, percent?

Notes: (Read and Write property)

macAddress as String

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** MAC address of wireless access point.

Notes: (Read and Write property)

name as String

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Name of current (or wanted?) network.

Example:

```
dim a as new WirelessMBS
dim info as WirelessInfoMBS=a.info
```

```
MsgBox info.name
```

Notes:

(currently returned with encoding set to ASCII)
(Read and Write property)

noise as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Noise level

Notes: (Read and Write property)

port_stat as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** HERMES_RID_PORTSTAT? (Uncertain about the meaning of this! 1=off? 2=connection bad? 3=AdHoc Create? 4=BSS (Client)? 5=BSS+OutOfRange?)

Notes: (Read and Write property)

power as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Power on flag.

Notes: (Read and Write property)

signal as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Signal level

Notes: (Read and Write property)

unknown7 as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Unknown data in the info structure.

Notes: (Read and Write property)

unknown9 as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Unknown data in the info structure.

Notes:

0=bad?, 1=ok?, 2=wrong key?

(Read and Write property)

2.10 class WindowsIPAddressMBS

class WindowsIPAddressMBS

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** An IP address.

2.10.1 Properties

IP as String

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** The IP as a string.

Notes:

Maximum 15 bytes long.
 e.g. "192.168.0.1"
 (Read and Write property)

Mask as String

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** The mask for the IP.
Notes:

e.g. "255.255.255.0"
 (Read and Write property)

2.11 class WirelessMBS

class WirelessMBS

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The class for Airport status and network search.

Notes:

This class is based on a private framework which Apple ships with Mac OS X. You use it on your own risk. Apple is not supporting this. (the plugins make still sure that it work for the current version. But we are not sure whether Apple won't change functionality in future versions)

Based on work from korben, jason and ragge:

Apple80211.h

This is the reverse engineered header for the Apple80211 private framework. The framework can be found at /System/Library/PrivateFrameworks/Apple80211.framework. Linking with Apple80211.framework requires CoreFoundation.framework and AppKit.framework.

Note that there is also information in the IORegistry, see
 ioreg -c AirPortDriver -w 0

Contributors:

korben - korben@cox.net

jason - catalyst@mac.com

ragge - ragge@nada.kth.se

Last updated by korben on 5/15/2002

ChangeLog:

2002-05-14 ragge

Changed argument types and count to procedures

Added WirelessScan

Changed name of unknown field to beaconInterval

Added error values and error return types

2002-05-15 korben

Combined ragge's changes with jason's

2002-05-17 korben

fixed adhoc and mangaged WINetworkInfoFlags per ragge's request

Added WirelessEncrypt and WirelessKey declarations

Updated WirelessJoinWEP and WirelessMakeIBSS comments regarding keys

2.11.1 Methods

BestChannel as integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** BestChannel is used to get the best channel for creating an adhoc network on.

Channels as integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Channels is used to get valid channels for creating an adhoc network.

Notes:

the return value contains a bit field of valid channels.

For example if 0x07FF is returned then bits 0 through 10 are set, which means channels 1 through 11 are valid.

Configure

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Calls some configure method. (functionality unknown)

Notes: Lasterror is set.

Enabled as boolean

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Whether Airport communication is enabled.

Notes:

setting this value will enable or disable Airport communication.

getting this value could have returned the Enabled state of Airport, but it seems to rather return the Power state.

(Read and Write computed property)

Encrypt(inNetworkPassword as string, use104bits as boolean) as string

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Encrypt is called from oinWEP and MakeIBSS to translate a string into a 40 or 104-bit Apple hashed WEP key.

Notes: Third argument is false for 40 bit key and true for 104 bit key.

GetAssociationInfo as dictionary

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Get information about the current association.

Notes:

The returned dictionary includes following entries:

keyData: Binary string buffer of the key (e.g., 32-octet PSK)

multiCipher: Integer 0 = none, 5 = CCMP?

channel: Integer

isIBSS: Boolean
 authMode: Integer; 2 = WPA-Personal; 3 = open, 129 = WPA2-Enterprise
 isWPA: Integer; 0 = not used, 1 = WPA, -128 == WPA2
 SSID: Binary string buffer of the SSID
 cipherMode: Integer; 0 = none, 4 = CCMP?

Lasterror is set.

GetCountAdhocList as integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Returns the number of networks in the adhoc list.

GetCountApList as integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Returns the number of networks in the ap list.

Example:

```
dim a as new WirelessMBS
dim info as WirelessInfoMBS=a.info

a.Scan false

MsgBox str(a.GetCountApList)+" networks found."
```

GetInfoASP as dictionary

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Get ASP information dictionary.

Notes:

The returned dictionary includes following entries:

Version: version number (e.g., 3.0)

Channel: channel (e.g., 1)

Vendor: vendor (e.g., 2)

GetItemAdhocList(index as integer) as WirelessNetworkInfoMBS

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Returns the item with the given index in the adhoc list.

Notes:

Index must be in range 0 to count-1.

Returns nil on any error.

GetItemApList(index as integer) as WirelessNetworkInfoMBS

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Returns the item with the given index in the ap list.

Notes:

Index must be in range 0 to count-1.

Returns nil on any error.

HCF_ GetInfo(RIDno as integer, bufsize as integer) as string

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Get information from the Hermes chip.

Notes:

RIDno is the Hermes RID number for the data to get, as
 & hFC01 - HERMES_ RID_ CNFOWNMACADDR
 & hFC02 - HERMES_ RID_ CNFDESIRESSID
 & hFDC1 - HERMES_ RID_ CURRENTCHANNEL
 and so on.

Don't know why, but 0xF100 - HERMES_ INQ_ TALLIES works here too, and a struct with the counters will be returned. (The data returned seems to be lagging, though, call twice for fresh data.)

Info as WirelessInfoMBS

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Returns info about the state of the current wireless connection.

Example:

```
dim a as new WirelessMBS
dim info as WirelessInfoMBS=a.info
```

```
MsgBox info.name
```

Notes: Returns nil on any error.

InterfaceDict as dictionary

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Returns a dictionary matching the wireless network interfaces to the drivers.

Notes:

The returned dictionary has a key,value pairs for wireless interfaces.

The key is the interface name and the value is the driver identifier, e.g., en1: com.apple.driver.AirPort.Atheros

Returns nil on any error.

IsAvailable as boolean

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Returns true if a wireless interface is available, false otherwise.

Example:

```
if WirelessMBS.IsAvailable then
MsgBox "Running"
end
```

Join(inNetworkName as string)

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Join is used to join a Wireless network.

Join8021x(SSID as string, Password as string)

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Joins a wireless network.

Notes: Lasterror is set.

JoinWEP(inNetworkName as string, inNetworkPassword as string)

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** JoinWEP is used to join an encrypted network.

Notes:

inNetworkName is the name of the network to join.

inNetworkPassword is the password/key of the network.

inNetworkPassword description:

- Passwords are just a string of any length, they will be hashed into a key.
- Keys should be passed as a hex string, optionally beginning with 0x, and must be either 10 digits for a 40bit key or 26 digits for a 104bit key, or an ascii/binary representation of the key, 5 or 13 bytes long.
- It can also be the empty string, meaning no encryption.

<http://kbase.info.apple.com/cgi-bin/WebObjects/kbase.woa/11/wa/query?searchMode=Expert& type=id& val=KC.106424>

MakeIBSS(inNetworkName as string, inNetworkPassword as string, inChannel as integer)

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** MakeIBSS is used to create a computer to computer adhoc wireless network.

Notes:

inNetworkName is the name of the network to create.
 inNetworkPassword is the password/key for the new network.
 inChannel is the wireless channel the network will use.

inNetworkPassword description:

- Passwords are just a string of any length, they will be hashed into a key.
- Keys should be passed as a hex string, optionally beginning with 0x, and must be either 10 digits for a 40bit key or 26 digits for a 104bit key.
- It can also be the empty string, meaning no encryption.

For more info see:

<http://kbase.info.apple.com/cgi-bin/WebObjects/kbase.woa/11/wa/query?searchMode=Expert& type=id& val=KC.106424>

Power as boolean

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Whether Airport is turned on or off.

Notes:

True if on.
 (Read and Write computed property)

Scan(stripDups as boolean)

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Scans for available wireless networks.

Example:

```
dim a as new WirelessMBS
dim info as WirelessInfoMBS=a.info
```

```
a.Scan false
```

```
MsgBox str(a.GetCountApList)+" networks found."
```

Notes:

It will create two lists of networks. The arrays hold WirelessNetworkInfoMBS objects.

Note: An adhoc network created on the computer the scan is running on will not be found. Info can be used to find info about a local adhoc network.

If stripDups is true, only one basestation for each SSID will be returned

ScanSplit(stripDups as boolean)

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Scans for available wireless networks.

Notes:

It will create two lists of managed and adhoc networks. The arrays hold WirelessNetworkInfoMBS objects.

Note: An adhoc network created on the computer the scan is running on will not be found. Info can be used to find info about a local adhoc network.

If stripDups is true, only one basestation for each SSID will be returned

2.11.2 Properties

Handle as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The internal handle for the Wireless Context reference.

Notes: (Read and Write property)

lasterror as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The last error code reported.

Notes: (Read and Write property)

2.11.3 Constants

Internal2Err = -2013261818

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the possible error values.

InternalErr = -2013261821

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the possible error values.

kWINetworkAdhocFlag = & h0002

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the flags for the network information.

kWINetworkCFPollable = & h0004

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the flags for the network information.

kWINetworkCFPollRequest = & h0008

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the flags for the network information.

kWINetworkChannelAgility = & h0080

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the flags for the network information.

kWINetworkEncryptedFlag = & h0010

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the flags for the network information.

kWINetworkManagedFlag = & h0001

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the flags for the network information.

kWINetworkPBCC = & h0040

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the flags for the network information.

kWINetworkReserved = & hFF00

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the flags for the network information.

kWINetworkShortPreamble = & h0020

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the flags for the network information.

NoIOServiceErr = -2013261822

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the possible error values.

NoPowerErr = -2013261813

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the possible error values.

OutOfMemErr = -2013261819

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the possible error values.

ParamErr = -2013261823

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the possible error values.

Unk4Err = -2013261820

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the possible error values.

Unk7Err = -2013261817

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the possible error values.

Unk8Err = -2013261816

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the possible error values.

Unk9Err = -2013261815

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the possible error values.

UnkaErr = -2013261814

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** One of the possible error values.

2.12 class WindowsEthernetMBS

class WindowsEthernetMBS

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** A list of the Windows Ethernet Interfaces.

Notes:

Requires Windows 98 or 2000 or newer.

Reports count=0 on Windows 95 and NT 4.0.

Only activate adapters are listed.

2.12.1 Methods

Item(index as integer) as WindowsEthernetAdapterMBS

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Returns the Ethernet Adapter object for the given index.

Notes: Index goes from 0 to count-1.

Update

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Updates the list.

Notes: Calls the Destructor and than the Constructor so the list is completely rebuilt.

2.12.2 Properties

Count as Integer

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** How much ethernet adapters are available.

Notes:

Note that there may be dummy or virtual adapters.
(Read and Write property)

2.13 class WindowsEthernetAdapterMBS

class WindowsEthernetAdapterMBS

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** A class for the properties of an ethernet adapter.

2.13.1 Methods

Gateway(index as integer) as WindowsIPAddressMBS

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Returns one of the IP address of the default gateway for this adapter.

Notes: Index goes from 0 to GatewayGount-1.

IP(index as integer) as WindowsIPAddressMBS

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Returns one of the IP addresses associated with this adapter.

Notes: Index goes from 0 to IPcount-1.

2.13.2 Properties

AdapterName as String

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Name of the adapter.

Notes: (Read and Write property)

Address as String

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Hardware address for the adapter.

Notes: (Read and Write property)

CurrentIpAddress as WindowsIPAddressMBS

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** The current IP from within the IP list.

Notes:

This field is currently normally nil, but in the future it may point to the current IP inside the IP list. But currently on Windows 2000 it seems to be always nil.
(Read and Write property)

Description as String

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Description for the adapter.

Notes: (Read and Write property)

DhcpEnabled as Boolean

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Specifies whether dynamic host configuration protocol (DHCP) is enabled for this adapter.

Notes: (Read and Write property)

DhcpServer as WindowsIPAddressMBS

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** IP address of the DHCP server for this adapter.

Notes: (Read and Write property)

Gatewaycount as Integer

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** How many gateway IPs are allocated for this adapter.

Notes: (Read and Write property)

HaveWins as Boolean

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Specifies whether this adapter uses Windows Internet Name Service (WINS).

Notes: (Read and Write property)

Index as Integer

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Adapter index.

Notes: (Read and Write property)

IPcount as Integer

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** How many IPs are allocated for this adapter.

Notes: (Read and Write property)

LeaseExpires as Integer

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Time when the current DHCP lease expires.

Notes: (Read and Write property)

LeaseObtained as Integer

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Time when the current DHCP lease was obtained.

Notes: (Read and Write property)

PrimaryWinsServer as WindowsIPAddressMBS

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** IP address of the primary WINS server.

Notes: (Read and Write property)

SecondaryWinsServer as WindowsIPAddressMBS

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** IP address of the secondary WINS server.

Notes: (Read and Write property)

Type as Integer

Plugin Version: 3.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Adapter type.

Notes:

The type must be of the following values:

MIB_IF_TYPE_OTHER	1
MIB_IF_TYPE_ETHERNET	6
MIB_IF_TYPE_TOKENRING	9
MIB_IF_TYPE_FDDI	15
MIB_IF_TYPE_PPP	23
MIB_IF_TYPE_LOOPBACK	24
MIB_IF_TYPE_SLIP	28

(Read and Write property)

2.14 class WirelessNetworkInfoMBS**class WirelessNetworkInfoMBS**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The data structure to hold information about a network.

2.14.1 Properties

beaconInterval as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** beacon interval in milliseconds

Notes: (Read and Write property)

channel as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Channel for the network.

Notes: (Read and Write property)

flags as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Flags for the network.
Notes:

See kWINetwork constants in the WirelessMBS class.
(Read and Write property)

macAddress as String

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** MAC address of the wireless access point.

Notes: (Read and Write property)

name as String

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The name of the network.

Notes:

(currently returned with encoding set to ASCII)
(Read and Write property)

nameLen as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The length of the name.

Notes: (Read and Write property)

noise as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Noise for the network. 0 for Adhoc.

Notes: (Read and Write property)

signal as Integer

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Signal strength of the network. 0 for Adhoc.

Notes: (Read and Write property)

2.15 class DNSServiceBrowseMBS

class DNSServiceBrowseMBS

Plugin Version: 5.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** A class for Apple's implementation of ZeroConfig named Bonjour.

Notes:

This class allows you to browse for services on the local network using Rendezvous.

This class requires Mac OS X 10.3 or Windows (with installed Bonjour support and the dnssd.dll) to work. Use DNSServiceDiscoveryBrowseMBS if you want to support Mac OS X 10.2.

common error codes:

kDNSServiceErr_NoError	= 0
kDNSServiceErr_Unknown	= -65537 (= & hFFFE FFFF)
kDNSServiceErr_NoSuchName	= -65538
kDNSServiceErr_NoMemory	= -65539
kDNSServiceErr_BadParam	= -65540
kDNSServiceErr_BadReference	= -65541
kDNSServiceErr_BadState	= -65542
kDNSServiceErr_BadFlags	= -65543
kDNSServiceErr_Unsupported	= -65544
kDNSServiceErr_NotInitialized	= -65545
kDNSServiceErr_AlreadyRegistered	= -65547
kDNSServiceErr_NameConflict	= -65548
kDNSServiceErr_Invalid	= -65549
kDNSServiceErr_Incompatible	= -65551 (client library incompatible with daemon)
kDNSServiceErr_BadinterfaceIndex	= -65552

Subclass of the `DNSServiceBaseMBS` class.

2.15.1 Methods

Browse(IntefaceIndex as integer, servicetype as string, domain as string) as boolean

Plugin Version: 5.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Asynchronously create a DNS Service browser to search for matching services in the local network.

Notes:

servicetype:

The type of service.

domain:

The domain in which to find the service.

Returns true if successful.

The `ServiceBrowse` event may be called till this browse request is closed.

You need to keep this object alive (keep a reference in a window, a module or your app class) so you can receive events.

In a perfect application you use `DNSServiceDomainEnumerationMBS` class to enumerate which domains you should use.

2.15.2 Events

ServiceBrowse(Flags as integer, InterfaceIndex as integer, ErrorCode as integer, ServiceName as string, RegType as string, Domain as string)

Plugin Version: 5.0 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** New service found or lost one.

Notes:

Values for flags:

<code>kDNSServiceFlagsAdd</code>	= 2
<code>kDNSServiceFlagsRemove</code>	= 0 (i.e. bit not set)
<code>kDNSServiceFlagsMoreComing</code>	= 1
<code>kDNSServiceFlagsFinished</code>	= 0 (i.e. bit not set)

`MoreComing` indicates to a `Browse` callback that another result is queued. Applications should not update their UI to display browse results when the `MoreComing` flag is set, instead deferring the update until the callback's flag is `Finished`.

Starting a new query from within this event on the same object can cause problems. Better you create a new instance for another query if you need one.

2.16 class DNSLookupMBS

class DNSLookupMBS

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for the result of a Lookup query.

Example:

```

dim d as DNSLookupMBS = DNSLookupMBS.LookupHostbyName("www.apple.com")

if d=nil then
MsgBox "No DNS Server available?"
else
MsgBox DNSLookupMBS.FormatIP(d.Address)
end if

```

2.16.1 Methods

Addresses(index as integer) as string

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the address with the given index for this host.

Example:

```

dim d as DNSLookupMBS = DNSLookupMBS.LookupHostbyName("www.apple.com")

if d=nil then
MsgBox "No DNS Server available?"
else
dim c as integer = d.AddressesCount-1
for i as integer = 0 to c
MsgBox d.FormatIP(d.Addresses(i))
next
end if

```

Notes: Index from 0 to count-1.

Aliases(index as integer) as string

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the aliasname for the given index.

Example:

```

dim d as DNSLookupMBS = DNSLookupMBS.LookupHostbyName("www.apple.com")

```

```

if d=nil then
MsgBox "No DNS Server available?"
else
dim c as integer = d.AliasCount-1
for i as integer = 0 to c
MsgBox d.Aliases(i)
next

end if

```

Notes: Index is from 0 to count-1.

FormatIP(ip as string) as string

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Formats an IP from binary representation into string representation.

Example:

```

dim s as string = DNSLookupMBS.ParseIPv4("65.66.67.68")

MsgBox s // in binary representation of that IP

dim t as string = DNSLookupMBS.FormatIP(s)

MsgBox t

```

Notes:

Formats both IPv4 and IPv6 IP addresses in binary representation into human readable text.
On Windows IPv6 is only supported on Windows Vista and newer.
Returns empty string on failure.

LookupHostbyAddress(HostAddressBinary as string) as DNSLookupMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Queries the DNS server for information about a host IP address.

Example:

```

dim l as DNSLookupMBS
dim a as string

a = DNSLookupMBS.ParseIPv4("17.254.0.91")

// try it...
l = DNSLookupMBS.LookupHostbyAddress(a)

// success...
MsgBox l.Name

```

Notes:

The address must be a 4 byte IP address like the ones returned by DNSLookupMBS.Address.
Returns nil on any error.

LookupHostbyAddressMT(HostAddressBinary as string) as DNSLookupMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Queries the DNS server for information about a host IP address.

Example:

```

dim l as DNSLookupMBS
dim a as string

a = DNSLookupMBS.ParseIPv4("17.254.0.91")

// try it...
l = DNSLookupMBS.LookupHostbyAddressMT(a)

// success...
MsgBox l.Name

```

Notes:

Same as LookupHostbyAddress, but with additional multithreading.
Must be called inside a Real Studio thread so time yields to main thread and you can keep the GUI running.

The address must be a 4 byte IP address like the ones returned by DNSLookupMBS.Address.

Returns nil on any error.

LookupHostbyName(HostName as string) as DNSLookupMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Queries the DNS server for information about a host name.

Example:

```
dim s as DNSLookupMBS = DNSLookupMBS.LookupHostbyName("www.monkeybreadsoftware.de")

dim c as integer = s.AddressesCount-1
for i as integer = 0 to c
dim ss as string = s.Addresses(i)
print str(s.FormatIP(ss))
next
```

Notes:

Will fail if you pass an IP as an host. (e.g. 10.20.30.40)

Returns nil on any error.

See also:

- 2.16.1 LookupHostbyName(HostName as string, AddressType as integer) as DNSLookupMBS 67

LookupHostbyName(HostName as string, AddressType as integer) as DNSLookupMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Queries the DNS server for information about a host name.

Example:

```
dim s as DNSLookupMBS = DNSLookupMBS.LookupHostbyName("ipv6.google.com", DNSLookupMBS.AddressTypeIPv6)

if s<>nil then
dim c as integer = s.AddressesCount-1
for i as integer = 0 to c
dim ss as string = s.Addresses(i)
print str(s.FormatIP(ss))
next
```

```
end if
```

```
s = DNSLookupMBS.LookupHostbyName("www.six.heise.de", DNSLookupMBS.AddressTypeIPv6)
```

```
if s<>nil then
dim c as integer = s.AddressesCount-1
for i as integer = 0 to c
dim ss as string = s.Addresses(i)
print str(s.FormatIP(ss))
next
end if
```

Notes:

Will fail if you pass an IP as an host. (e.g. 10.20.30.40)
Returns nil on any error.

IPv6 on Windows does not work with this method.
See also:

- 2.16.1 LookupHostbyName(HostName as string) as DNSLookupMBS

67

LookupHostbyNameMT(HostName as string) as DNSLookupMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Queries the DNS server for information about a host name.

Example:

```
dim s as DNSLookupMBS = DNSLookupMBS.LookupHostbyNameMT("www.monkeybreadsoftware.de")

dim c as integer = s.AddressesCount-1
for i as integer = 0 to c
dim ss as string = s.Addresses(i)
print str(s.FormatIP(ss))
next
```

Notes:

Same as LookupHostbyName, but with additional multithreading.

Must be called inside a Real Studio thread so time yields to main thread and you can keep the GUI running.

Will fail if you pass an IP as a host. (e.g. 10.20.30.40)

Returns nil on any error.

See also:

- 2.16.1 LookupHostbyNameMT(HostName as string, AddressType as integer) as DNSLookupMBS 69

LookupHostbyNameMT(HostName as string, AddressType as integer) as DNSLookupMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Queries the DNS server for information about a host name.

Example:

```
dim s as DNSLookupMBS = DNSLookupMBS.LookupHostbyNameMT("ipv6.google.com", DNSLookupMBS.AddressTypeIPv6)
```

```
if s<>nil then
dim c as integer = s.AddressesCount-1
for i as integer = 0 to c
dim ss as string = s.Addresses(i)
print str(s.FormatIP(ss))
next
end if
```

```
s = DNSLookupMBS.LookupHostbyNameMT("www.six.heise.de", DNSLookupMBS.AddressTypeIPv6)
```

```
if s<>nil then
dim c as integer = s.AddressesCount-1
for i as integer = 0 to c
dim ss as string = s.Addresses(i)
print str(s.FormatIP(ss))
next
end if
```

Notes:

Same as LookupHostbyName, but with additional multithreading.

Must be called inside a Real Studio thread so time yields to main thread and you can keep the GUI running.

Will fail if you pass an IP as an host. (e.g. 10.20.30.40)
Returns nil on any error.

IPv6 on Windows does not work with this method.
See also:

- 2.16.1 LookupHostNameMT(HostName as string) as DNSLookupMBS

68

ParseIPv4(ip as string) as string

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Parses an IP address in IPv4 format.

Example:

```
dim s as string = DNSLookupMBS.ParseIPv4("65.66.67.68")
```

```
MsgBox s // in binary representation of that IP
```

```
dim t as string = DNSLookupMBS.FormatIP(s)
```

```
MsgBox t
```

Notes:

Returns empty string on failure.
Works only for IPv4 strings.

ParseIPv6(ip as string) as string

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Parses an IP address in IPv6 format.

Example:

```
dim s as string = DNSLookupMBS.ParseIPv6("2001:0db8:85a3:08d3:1319:8a2e:0370:7344")
```

```
MsgBox s // in binary representation of that IP
```

```
dim t as string = DNSLookupMBS.FormatIP(s)
```

MsgBox t

Notes:

Returns empty string on failure.
Works only for IPv6 strings.

2.16.2 Properties

Address as String

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The primary address of the host.

Example:

```
dim d as DNSLookupMBS = DNSLookupMBS.LookupHostbyName("www.apple.com")
if d=nil then
  MsgBox "No DNS Server available?"
else
  MsgBox DNSLookupMBS.FormatIP(d.Address)
end if
```

Notes:

On TCP/IP based system the address is 4 bytes long binary string.
(Read and Write property)

AddressesCount as Integer

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of addresses for this host.

Example:

```

dim d as DNSLookupMBS = DNSLookupMBS.LookupHostbyName("www.apple.com")

if d=nil then
  MsgBox "No DNS Server available?"
else
  MsgBox str(d.AddressesCount)
end if

```

Notes: (Read and Write property)

AddressType as Integer

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The type of the address format.

Example:

```

dim d as DNSLookupMBS = DNSLookupMBS.LookupHostbyName("www.apple.com")

if d=nil then
  MsgBox "No DNS Server available?"
else
  MsgBox str(d.AddressType) // shows 2 for IPv4
end if

```

Notes:

2 for IPv2

10 for IPv10

(Read and Write property)

AliasCount as Integer

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of aliases for this host.

Example:

```

dim d as DNSLookupMBS = DNSLookupMBS.LookupHostbyName("www.apple.com")

if d=nil then
MsgBox "No DNS Server available?"
else
MsgBox str(d.AliasCount)
end if

```

Notes: (Read and Write property)

Name as String

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The name of the host.

Example:

```

dim d as DNSLookupMBS = DNSLookupMBS.LookupHostbyName("www.apple.com")

if d=nil then
MsgBox "No DNS Server available?"
else
MsgBox d.Name
end if

```

Notes: (Read and Write property)

2.16.3 Constants

AddressTypeIPv4 = 2

Plugin Version: 10.4 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the address types.

Notes: An IPv4 address.

AddressTypeIPv6 = 10

Plugin Version: 10.4 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the address types.

Notes: An IPv6 address.

2.17 class DNSServiceBaseMBS**class DNSServiceBaseMBS**

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The base DNS Service class.

2.17.1 Methods**Available as boolean**

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Whether Bonjour (zeroconfig) is installed and useable.

Notes: If this function returns true, all the needed stuff is installed and the function have been loaded.

Close

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Stops the current request and releases all used memory.

Notes: This is done automatically for you by the destructor.

ConstructFullName(Service as string, regtype as string, domain as string) as string

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Concatenate a three-part domain name (as returned by the above callbacks) into a properly-escaped full domain name.

Notes:

Note that events in the above functions ALREADY ESCAPE strings where necessary.

service: The service name - any dots or backslashes must NOT be escaped. May be "" (to construct a PTR record name, e.g. "_ftp._tcp.apple.com").

regtype: The service type followed by the protocol, separated by a dot (e.g. "_ftp._tcp").

domain: The domain name, e.g. "apple.com.". Literal dots or backslashes, if any, must be escaped, e.g. "1st\Floor.apple.com."

Returns 0 on success, -1 on error.

GetDaemonVersion as integer

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Queries the version of the daemon.

Notes: Value is zero on any error.

Running as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** True if there is a running request.

2.17.2 Properties

Handle as Integer

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The handle for this request.

Notes: (Read and Write property)

Lasterror as Integer

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The last error code reported.

Notes:

0 if the function succeeded.

-1 if the function used was not available.

See the kError constants for possible errors.

(Read and Write property)

2.17.3 Constants

kClassIN = 1

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** The DNS class for Internet services.

kErrorAlreadyRegistered = -65547

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorBadFlags = -65543

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorBadInterfaceIndex = -65552

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorBadParam = -65540

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorBadReference = -65541

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorBadState = -65542

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorBadTime = -65559

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorDoubleNAT = -65558

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorFirewall = -65550

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorIncompatible = -65551

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorInvalid = -65549

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorNameConflict = -65548

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorNATTraversal = -65557

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorNoAuth = -65555

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorNoError = 0

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorNoMemory = -65539

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorNoSuchKey = -65556

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorNoSuchName = -65538

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorNoSuchRecord = -65554

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorNotInitialized = -65545

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorRefused = -65553

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorUnknown = -65537

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kErrorUnsupported = -65544

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the error constants.

kFlagsAdd = 2

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** A flag for domain enumeration and browse/query events.

Notes: "Default" applies only to enumeration and is only valid in conjunction with "Add". An enumeration callback with the "Add" flag NOT set indicates a "Remove", i.e. the domain is no longer valid.

kFlagsAllowRemoteQuery = & h200

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Flag for creating a record for which we will answer remote queries (queries from hosts more than one hop away; hosts not directly connected to the local link).

kFlagsBrowseDomains = & h40

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Flags for specifying domain enumeration type in DNSServiceEnumerateDomains.

Notes: BrowseDomains enumerates domains recommended for browsing, RegistrationDomains enumerates domains recommended for registration.

kFlagsDefault = 4

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** A flag for domain enumeration and browse/query events.

Notes: "Default" applies only to enumeration and is only valid in conjunction with "Add". An enumeration callback with the "Add" flag NOT set indicates a "Remove", i.e. the domain is no longer valid.

kFlagsForceMulticast = & h400

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the flags.

Notes: Flag for signifying that a query or registration should be performed exclusively via multicast DNS, even for a name in a domain (e.g. foo.apple.com.) that would normally imply unicast DNS.

kFlagsLongLivedQuery = & h100

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Flag for creating a long-lived unicast query for the QueryRecord call.

kFlagsMoreComing = 1

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the flags used for the events.

Notes: MoreComing indicates to a callback that at least one more result is queued and will be delivered following immediately after this one. Applications should not update their UI to display browse results when the MoreComing flag is set, because this would result in a great deal of ugly flickering on the screen. Applications should instead wait until until MoreComing is not set, and then update their UI. When MoreComing is not set, that doesn't mean there will be no more answers EVER, just that there are no more answers immediately available right now at this instant. If more answers become available in the future they will be delivered as usual.

kFlagsNoAutoRename = 8

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** The flag for specifying renaming behavior on name conflict when registering non-shared records.

Notes: By default, name conflicts are automatically handled by renaming the service. NoAutoRename

overrides this behavior - with this flag set, name conflicts will result in a callback. The NoAutorename flag is only valid if a name is explicitly specified when registering a service (i.e. the default name is not used.)

kFlagsRegistrationDomains = & h80

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Flags for specifying domain enumeration type in DNSServiceEnumerateDomains.

Notes: BrowseDomains enumerates domains recommended for browsing, RegistrationDomains enumerates domains recommended for registration.

kFlagsShared = & h10

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Flag for registering individual records on a connected DNSService.

Notes: Shared indicates that there may be multiple records with this name on the network (e.g. PTR records). Unique indicates that the record's name is to be unique on the network (e.g. SRV records).

kFlagsUnique = & h20

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Flag for registering individual records on a connected DNSService.

Notes: Shared indicates that there may be multiple records with this name on the network (e.g. PTR records). Unique indicates that the record's name is to be unique on the network (e.g. SRV records).

kInterfaceIndexAny = 0

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** The constant to use for interface index to target any interface.

kInterfaceIndexLocalOnly = -1

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** The constant to use for interface index to target the local interface.

kMaxDomainName = 1005

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Maximum length, in bytes, of a domain name represented as an *escaped* C-String including the final trailing dot, and the C-String terminating NULL at the end.

kMaxServiceName = 64

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Maximum length, in bytes, of a service name represented as a literal C-String, including the terminating NULL at the end.

kTypeA = 1

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Host address.

kTypeA6 = 38

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: IPv6 address (deprecates AAAA)

kTypeAAAA = 28

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Ip6 Address.

kTypeAFSDB = 18

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: AFS cell database.

kTypeANY = 255

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Wildcard match.

kTypeATMA = 34

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: ATM Address

kTypeAXFR = 252

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Transfer zone of authority.

kTypeCERT = 37

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Certification record.

kTypeCNAME = 5

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Canonical name.

kTypeDNAME = 39

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Non-terminal DNAME (for IPv6)

kTypeEID = 31

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Endpoint identifier.

kTypeGPOS = 27

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Geographical position (withdrawn).

kTypeHINFO = 13

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Host information.

kTypeISDN = 20

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: ISDN calling address.

kTypeIXFR = 251

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Incremental zone transfer.

kTypeKEY = 25

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Security key.

kTypeKX = 36

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Key Exchange

kTypeLOC = 29

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Location Information.

kTypeMAILA = 254

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Transfer mail agent records.

kTypeMAILB = 253

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Transfer mailbox records.

kTypeMB = 7

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Mailbox domain name.

kTypeMD = 3

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Mail destination.

kTypeMF = 4

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Mail forwarder.

kTypeMG = 8

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Mail group member.

kTypeMINFO = 14

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Mailbox information.

kTypeMR = 9

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Mail rename name.

kTypeMX = 15

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Mail routing information.

kTypeNAPTR = 35

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Naming Authority PoinTeR

kTypeNIMLOC = 32

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Nimrod Locator.

kTypeNS = 2

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Authoritative server.

kTypeNSAP = 22

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: NSAP address.

kTypeNSAP_PTR = 23

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Reverse NSAP lookup (deprecated).

kTypeNULL = 10

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Null resource record.

kTypeNXT = 30

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Next domain (security).

kTypeOPT = 41

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: EDNS0 option (meta-RR)

kTypePTR = 12

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Domain name pointer.

kTypePX = 26

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: X.400 mail mapping.

kTypeRP = 17

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Responsible person.

kTypeRT = 21

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Router.

kTypeSIG = 24

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Security signature.

kTypeSINK = 40

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Kitchen sink (experimentatl)

kTypeSOA = 6

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Start of authority zone.

kTypeSRV = 33

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Server Selection.

kTypeTKKEY = 249

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Transaction key.

kTypeTSIG = 250

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Transaction signature.

kTypeTXT = 16

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: One or more text strings.

kTypeWKS = 11

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: Well known service.

kTypeX25 = 19

Plugin Version: 10.1 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** One of the DNS Service type constants.

Notes: X-25 calling address.

2.18 class DarwinPingMBS

class DarwinPingMBS

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** A class to do a ping on Mac OS X.

Notes: This class has still issues, so you may prefer to run ping in the shell class.

2.18.1 Methods

Ping(HostToPing as string, TimeOutMS as integer, TimeToLife as integer) as integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Starts a synchron ping.

Notes:

This method does not return until the time passed or the ping response was received.
Return value is a platform dependend error code. 0 is successful.

SimplePing(HostToPing as string, NumberOfPacketsToSend as integer, PingTimeoutInSeconds as integer, ReturnImmediatelyAfterReply as integer) as integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Starts an asynchron ping.

Notes:

This method will call the events and return when it finished.
This method can be called in a thread.
Return value is a platform dependend error code. 0 is successful.

2.18.2 Properties

HostToPing as String

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The value used in the Ping command.

Notes: (Read only property)

NumberOfPacketsToSend as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The value used in the Ping method.

Notes: (Read only property)

PingTimeoutInSeconds as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The ping timeout to use.

Notes:

0 for default.

(Read only property)

ReturnImmediatelyAfterReply as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The value used in the Ping method.

Notes: (Read only property)

TimeToLife as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The time to life value for the ping packets.

Notes:

a value between 0 and 255.
(Read only property)

2.18.3 Events

AddressResolved(ip as string)

Plugin Version: 4.2 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** The IP for the domain name was resolved.

Finished(NumberPacketsSent as integer, NumberPacketsReceived as integer)

Plugin Version: 4.2 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** The pings have finished.

NextPing(PacketSequenceNumber as integer)

Plugin Version: 4.2 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** We are starting the next ping now.

Response(PacketSequenceNumber as integer, ttl as integer, RoundTripTimeInMS as double)

Plugin Version: 4.2 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** Got a response for the given packet.

SentError(PacketSequenceNumber as integer)

Plugin Version: 4.2 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** A packet could not be sent.

SentSuccess(PacketSequenceNumber as integer)

Plugin Version: 4.2 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** The packet has sent successfull.

Timeout(PacketSequenceNumber as integer)

Plugin Version: 4.2 Console & Web: No Mac: Yes, Win: No, Linux: No, . **Function:** A timeout was reached for the given packet.

Chapter 3

List of all classes

• DarwinPingMBS	93
• DNSLookupMBS	63
• DNSServiceBaseMBS	74
• DNSServiceBrowseMBS	61
• DNSServiceDomainEnumerationMBS	31
• DNSServiceRegisterMBS	36
• DNSServiceRegisterRecordMBS	33
• DNSServiceResolveMBS	34
• FTTParseMBS	20
• PacketSocketMBS	25
• TXTRecordMBS	11
• WindowsEthernetAdapterMBS	55
• WindowsEthernetMBS	54
• WindowsIPAddressMBS	41
• WirelessInfoMBS	39
• WirelessMBS	42
• WirelessNetworkInfoMBS	59

Chapter 4

List of all global methods

- 2.4 ClearOptionsMBS(extends s as SocketCore) 27
- 2.4 DNSAddressToNameIPv6MBS(HostAddress as string) as string 27
- 2.4 DNSAddressToNameMBS(HostAddress as string) as string 27
- 2.4 DNSNameToAddressIPv6MBS(HostName as string) as string 27
- 2.4 DNSNameToAddressMBS(HostName as string) as string 28
- 2.4 OptionMutliCastTTLMBS(extends s as SocketCore) as integer 28
- 2.4 OptionMutliCastTTLMBS(extends s as SocketCore, assigns value as integer) 28
- 2.4 OptionTOSMBS(extends s as SocketCore) as integer 29
- 2.4 OptionTOSMBS(extends s as SocketCore, assigns value as integer) 29
- 2.4 OptionTTLMBS(extends s as SocketCore) as integer 29
- 2.4 OptionTTLMBS(extends s as SocketCore, assigns value as integer) 30